

REMARKS

Prior to this Reply, Claims 1-29 were pending. Through this Reply, Claims 12, 19-24 and 26-29 have been amended. No claims have been added or cancelled. Accordingly, Claims 1-29 are now at issue in the present case.

I. Allowable Subject Matter

The Examiner objected to Claims 4-16, 20-24 and 27-29 as being dependent upon a rejected base claim. However, the Examiner indicated that such claims would be allowable if rewritten in independent form to include all of the limitations of their corresponding base claims and any intervening claims.

Applicant thanks the Examiner for her indication that the aforementioned claims contain allowable subject matter. Instead of rewriting the claims in independent form, Applicant offers the arguments presented below.

II. Rejection of Claims 1-3, 17-19, 25 and 26 Under 35 U.S.C. § 103(a)

The Examiner rejected Claims 1-3, 17-19, 25 and 26 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,373,512 to Brady (hereinafter "Brady"). The rejection of such claims is respectfully traversed.

A. Brief Discussion of Brady

Brady is directed to a method and means for generating parity for an array of redundant storage devices, such as disk files (RAID). A memory controller is provided to reduce data transfers, and hence bus bandwidth, minimizing movement of data to and from a dedicated parity

generator. Specifically, Brady teaches a circuit (FIG. 2), wherein data blocks are transferred to a memory 21 (e.g., memory 21a) via a channel 20 and bus 24. The memory 21a includes a memory array 23 and a memory controller 22 having a parity generator 34 (FIG. 3). For $n+1$ disk files 25, Brady desires to write n data blocks on n different disk files, calculate parity, and write the parity, as calculated for the said n blocks, on the $n+1$ th disk file. In Col. 3, line 31 to Col. 4, line 11 and in FIG. 2, Brady describes that the n data blocks are supplied via the channel interface 20 and bus 24, and are directed serially to each memory 21 (i.e., 21a and 21b). The memory 21a is selected to calculate parity, and responds to a control signal from bus 24, to perform an XOR to the contents of a portion or location P of the memory array 23 allocated for parity calculation. As the data blocks continue to arrive via channel interface 20, they are XOR'd one-by-one to the partial parity as calculated at said allocated location P. When the last of the n blocks is transmitted, parity is available in memory 21a and is written in memory array 23 and also via disk controller 26 to the appropriate disk file 25'.

B. Brady Fails to Disclose All of the Limitations of the Claims

Applicant believes that Brady fails to disclose every limitation of the claims.

Accordingly, Applicant believes that the claims are patentably distinguishable from Brady.

As the Examiner admitted in the Office Action, Brady does not teach or suggest “maintaining a checksum list comprising a plurality of entries corresponding to the data segments stored in the buffer memory, each entry for storing a checksum value for a corresponding data segment stored in the buffer memory” as required by Claim 1. Furthermore, Brady does not teach or suggest storing checksum values in a checksum list. Even further, Brady does not teach or suggest, for each data segment retrieved from the storage device, “calculating a

checksum value for that data segment using a checksum circuit for performing a checksum function” and “storing the checksum value in an entry in the checksum list.”

In other words, a checksum is calculated for each individual data segment using the checksum circuit. And, the checksum for each data segment is individually maintained.

By contrast, Brady teaches that as the data blocks continue to arrive via channel interface 20, they will be XOR'd one-by-one to the partial parity as calculated at a parity allocated location P. When the last of the n blocks is transmitted, parity is available in memory 21a and can be written in memory array 23 and also via disk controller 26 to the appropriate disk file 25' (Col. 3, lines 56-63).

This has nothing to do with the invention as claimed in Claim 1, which includes the steps of, for each data segment retrieved from the storage device, “calculating a checksum value for that data segment using a checksum circuit for performing a checksum function” and “storing the checksum value in an entry in the checksum list.” In contrast, Brady generates a total parity for the n data blocks, and does not maintain a separate parity for each block.

As Brady specifically states, the n data blocks are stored in location D of the array 23 of memory 21b in response to control signals specifying normal storage. Once parity is calculated in memory 21a, it will be sent, in response to a control signal supplied from bus 24, onto the bus and gated out to disk controller 26. Controller 26 then directs the parity to the appropriate disk 25' (Col. 3, line 66 to Col. 4, line 3).

As the Examiner correctly stated, Brady does not teach storing checksum values in a checksum list. However, the Examiner has modified Brady to meet the limitations of the present invention by adding a series of limitations, described on pages 3-4 of the Office Action, that are not taught or suggested by Brady, and are not obvious to one of ordinary skill in the art. For

example, Claim 1 includes the limitation of: “maintaining a checksum list comprising a plurality of entries corresponding to the data segments stored in the buffer memory, each entry for storing a checksum value for a corresponding data segment stored in the buffer memory.” No such limitation is taught or suggested by Brady.

The Office Action essentially states that Brady teaches a write parity mode wherein one set of control signals identify the location P at which parity calculation is to be updated and XOR write operation is to be performed. Another set of control signals identify the location D from which data is to be read into the write buffer. The write buffer data from locations P and D is synchronized, and moved to the parity generator. Once parity is calculated, the data and memory array parity are stored in the memory array and stored in location P.

The Examiner states that it would have been obvious to choose to store Brady’s partial parity into a memory which also stores data words, or to store Brady’s partial parity into a separate memory. The Examiner reasoned that one skilled in the art would have been motivated to make such a change because such a choice would not change the content of Brady’s partial parity.

Applicant respectfully disagrees. Brady’s system operates as described, and does not need a checksum list of any sort. Maintaining a checksum list is not in anyway useful to Brady. Further, it is respectfully submitted that there is no reason, suggestion, or motivation in Brady to maintain a parity for each block. Rather, Brady uses a running parity (partial parity) in the process of generating total parity for a sequence of n data blocks. Brady does not maintain a separate checksum for each data block. Maintaining a separate parity for each data block is not useful to Brady and, indeed, may be burdensome to Brady. Brady needs a total parity for the n blocks, which Brady calculates using a running parity as it processes the blocks.

In Col. 4, line 20 to Col. 5, line 29, Brady teaches that, in operation, to write data into memory in normal write mode, a control signal is transmitted from data bus 24 via line 38 to control device 36. This control signal causes device 36, via line 39, to set disk parity generator 34 to a pass-through mode. As such, data is routed from the bus 24 to the location D in memory array 23 of memory 21. After the data is thus transmitted, control device 36 will condition itself to a wait state and await the next instruction. To read data in normal read mode, a control signal from bus 24 via line 38 conditions control device 36 to select the location D in the memory array 23, and to activate switch 37 to connect line 40 to line 41. As a result, the data to be read is gated from location D in memory array 23 via read path 31 to the ECC check/correction unit 33 to be corrected if necessary. The correct data is then transmitted from location D via switch 37 from line 40 via line 41.

In the “write parity” mode, data is written from location D to location P in memory array 23 and logically combined with the existing contents of said memory location to update the parity calculation P. This combining step is recursively repeated until the final parity calculation is generated. In FIG. 3, wherein locations D and P are in the same memory array 23, two sets of control signals are transmitted sequentially via line 38 to control device 36. One set of control signals identifies the location P at which parity calculation is to be updated and that an XOR write operation is to be performed. The other set of control signals identifies the location D from which data is to be read into the write buffer 35.

As such, Data is written into write buffer 35 from locations P and D, and moved in parallel from write buffer 35 over two paths into disk parity generator 34. Generator 34 will have been conditioned by control device 36 via line 39 to XOR the data words together bit-by-bit, then transmit the XOR'd data word to ECC generator 32, which will have been conditioned

via line 47 to calculate ECC parity for the memory array 23. Once parity is calculated, the data and memory array parity are transmitted via write path 30 to memory array 23 and there stored in location P. This process is repeated recursively until the required number of words from locations D and P in memory array 23 have been XOR'd together to calculate partial parity for one block. This process is then repeated recursively for successive blocks, updating the parity calculation stored at location P until the final parity calculation is completed. Memory 21 resumes a wait state awaiting the next instruction after each partial parity calculation.

As such, Brady's partial parity at each point in time, is simply a running parity for all the blocks processed up till that point in time. Brady does not maintain a separate checksum for each block as taught by the present invention. In one example, according to the present invention, the checksum list is useful "for transferring packets of data out of the buffer memory ..." wherein "... a checksum value can be calculated for data in each packet based on one or more checksum values stored in the checksum list" (Claim 1).

For at least the above reasons, Applicant submits that independent Claim 1 is patentably distinguishable from the cited reference. Further, for at least the same reasons, Applicant submits that all of the claims that depend from Claim 1 are likewise patentably distinguishable from the cited references. Even further, for reasons similar to those presented with respect to Claim 1, Applicant submits that Claims 17 and 25, and the claims that depend therefrom, are likewise patentably distinguishable from Brady.

Applicant submits that Claim 2 is patentably distinguishable from Brady for reasons in addition to those set forth above. Specifically, Applicant believes that Brady fails to disclose or suggest the step of "calculating the checksum for said data segment using the checksum circuit as the data segment is transferred into the buffer memory" as required by Claim 2. As discussed

above, the parity calculation in Brady is performed after the data is read into memory 23 in the “write parity” mode where data from locations P and D in memory 23 are moved to the write buffer 35 and then moved to parity generator 34 to generate parity data, and then the data and parity are stored back in memory 23. Therefore, Applicant submits that Claim 2 is patentably distinguishable from Brady.

Applicant submits that Claims 3 and 18 are patentably distinguishable from the cited reference for reasons in addition to those set forth above with respect to Claims 1 and 17. Regarding Claim 3, Brady does not teach or suggest the steps of: “building packets of data for transfer out of the buffer memory” and “providing a checksum value for data in each packet based on one or more checksum values stored in the checksum list.” Regarding Claim 18, Brady does not teach or suggest: “a logic circuit for calculating a checksum value for a data segment,” “means for locating an entry in the checksum table corresponding to that data segment,” and “means for storing the checksum value in the located entry in the checksum list.”

Applicant submits that Claims 19 and 26 are patentably distinguishable from Brady for reasons in addition to those set forth above with respect to Claims 1 and 17. Specifically, Applicant believes that the cited reference fails to disclose or suggest “a microcontroller that builds packets of data for transfer out of the buffer memory, and provides a checksum value for data in each packet based on one or more checksum values stored in the checksum list” as required by Claim 19. Further, Applicant believes that the cited reference fail to disclose or suggest a checksum system comprising “a processor configured for building packets of data for transfer out of the buffer memory, and for providing a checksum value for data in each packet based on one or more checksum values stored in the checksum list” as required by Claim 26.

III. Amendments to Claims 12, 19-24 and 26-29

Claims 12, 19-24 and 26-29 have been amended to clarify such claims. Importantly, such claims have not been amended in response to any prior art cited by or submitted to the U.S. Patent Office. Furthermore, no new matter has been added.

IV. Additional Claim Fees

In determining whether additional claim fees are due, reference is made to the Fee Calculation Table (below).

Fee Calculation Table

	Claims Remaining After Amendment		Highest Number Previously Paid For	Present Extra	Rate	Additional Fee
Total (37 CFR 1.16(c))	29	Minus	29	= 0	x \$18 =	\$ 0.00
Independent (37 CFR 1.16(b))	3	Minus	3	= 0	x \$84 =	\$ 0.00

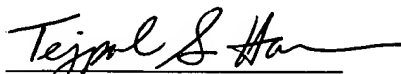
As set forth in the Fee Calculation Table (above), Applicant previously paid claim fees for twenty-nine (29) total claims and for three (3) independent claims. Accordingly, Applicant believes that no additional claims fees are due. Nevertheless, Applicant hereby authorizes the Commissioner to charge Deposit Account No. 50-2198 for any fee deficiencies associated with filing this paper.

V. Conclusion

Applicant believes that the application appears to be in form for allowance. Accordingly, reconsideration and allowance thereof is respectfully requested.

The Examiner is invited to contact the undersigned at the below-listed telephone number regarding any matters relating to the present application.

Respectfully submitted,



Tejpal S. Hansra
Registration No. 38,172
Hansra Patent Services
4525 Glen Meadows Place
Bellingham, WA 98226
(360) 527-1400

Date: SEPT. 25, 2003

RECEIVED

OCT 01 2003

OFFICE OF PETITIONS